

# How to Find Common Values Between Two Lists in Excel

Authored by  
**stats writer**

February 3, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Find Common Values Between Two Lists in Excel*.  
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=129292>

Identifying the intersection of two data sets--a common requirement in data analysis--can be efficiently managed within Microsoft Excel. The intersection represents the collection of elements that are present in both lists simultaneously. While simple list comparisons might rely on basic logical functions, a highly effective and standard methodology utilizes a combination of the **IF**, **ISERROR**, and **MATCH** functions. This powerful combination allows us to systematically compare entries across two separate columns, returning only the values that exist in both ranges. This technique is superior for extracting the actual common values, ensuring that data integrity and clarity are maintained throughout the analytical process.

## What is the method to find the intersection of two column lists in Excel?

### The Core Formula for Identifying Common Values

To accurately determine which elements are shared between two column lists, we employ a sophisticated conditional formula. This formula leverages the positional checking capability of **MATCH**, handles errors gracefully using **ISERROR**, and provides a clear output using the **IF** function. This combination ensures that the output column only displays values that successfully match an entry in the comparison column.

The standard syntax for this method, assuming your primary list is in Column A and your comparison list is in Column B, is as follows. Note the use of absolute references (\$) for the lookup range, which is critical for consistent dragging of the formula:

```
=IF(ISERROR(MATCH(A2,$B$2:$B$11,0)),"",A2)
```

In this construction, the formula is initially entered into the first cell of a new output column (e.g., D2). It then checks the value in **A2** against the entire range **B2:B11**. The use of absolute cell references (**\$B\$2:\$B\$11**) ensures that as the formula is copied down to cells like D3, D4, and so forth, the comparison range remains fixed, preventing calculation errors and ensuring that the entire comparison list is always utilized. This approach efficiently identifies all common values between the range **A2:A11** and **B2:B11**, preparing the analyst for the next step: practical implementation.

### Practical Example: Finding the Intersection of Two Basketball Team Lists

To illustrate the efficiency of this formula, let us consider a practical scenario involving two separate lists of professional basketball teams. Imagine we have one list compiled from last season's Western Conference rankings (Column A) and another list representing this season's

projected top contenders (Column B). Our goal is to identify exactly which teams are common to both lists, thus forming the set intersection. This process requires meticulous data comparison, which is automated by our chosen formula.

The initial data setup is presented below. Column A contains the first list (A2:A11), and Column B contains the second list (B2:B11).

	A	B	C	D	E
1	<b>Team 1</b>	<b>Team 2</b>			
2	Mavs	Warriors			
3	Spurs	Kings			
4	Rockets	Celtics			
5	Kings	Nuggets			
6	Warriors	Mavs			
7	Nets	Hawks			
8	Lakers	Magic			
9	Thunder	Blazers			
10	Blazers	Rockets			
11	Jazz	Knicks			
12					
13					
14					
15					

The objective is clear: we seek the subset of team names that are duplicated across both the A and B ranges. This extraction process starts by initializing the formula in an adjacent column, such as **D**, beginning in cell **D2**. The formula must reference the first item in the primary list (A2) and the entirety of the comparison list (B2:B11) using the required absolute references:

**=IF(ISERROR(MATCH(A2,\$B\$2:\$B\$11,0)),"",A2)**

Once entered into **D2**, this powerful formula performs the crucial check. If the value of **A2** (the first team name) is successfully located within the comparison range B, the formula proceeds to display that team name in cell D2. If, however, the team name is unique to list A, the **MATCH** function fails, triggering the **ISERROR** condition, which then results in a blank cell. The final step involves executing the fill handle function: clicking and dragging the formula from **D2** downwards, replicating the logic across the entire length of List A (down to D11).

Upon completing the drag-down operation, the resulting Column D clearly isolates the common

entries. This visual representation immediately verifies the intersection, providing a filtered list of teams that satisfy the criteria of being present in both the Western Conference rankings and the projected contenders list. The resulting table is shown below, confirming the successful execution of the intersection logic:

	A	B	C	D	E	F
1	<b>Team 1</b>	<b>Team 2</b>		<b>Intersection of Teams</b>		
2	Mavs	Warriors		Mavs		
3	Spurs	Kings				
4	Rockets	Celtics		Rockets		
5	Kings	Nuggets		Kings		
6	Warriors	Mavs		Warriors		
7	Nets	Hawks				
8	Lakers	Magic				
9	Thunder	Blazers				
10	Blazers	Rockets		Blazers		
11	Jazz	Knicks				
12						
13						
14						
15						

As clearly demonstrated by the output in Column D, only the team names that appeared in both lists A and B are now visible. This process confirms the intersection set. Specifically, the following five teams were successfully identified as common members of both data sets:

**Mavs**

**Rockets**

**Kings**

**Warriors**

**Blazers**

These results form the definitive intersection between the two initial column lists, validating the effectiveness and reliability of the IF/ISERROR/MATCH formula combination for complex data filtering tasks in Excel.

## Detailed Breakdown: Deconstructing the IF/ISERROR/MATCH Formula

Understanding the operational sequence of the components within the primary formula,

`=IF(ISERROR(MATCH(A2,$B$2:$B$11,0)), "", A2)`, is crucial for mastering its application. This formula is a powerful example of nested logic, where the output of one function serves as the input for the next. The entire expression is structured around three core Excel functions, each playing a vital role in the comparison and filtering process.

The process initiates with the **MATCH** function, which is responsible for the actual comparison. The structure `MATCH(A2,$B$2:$B$11,0)` instructs Excel to search for the exact value contained in cell **A2** within the specified array (the comparison list, **\$B\$2:\$B\$11**). The final argument, `0`, is critical as it specifies an exact match requirement. If **A2** is successfully found within the B range, the function returns the relative row position of that match within the array. Crucially, if the value in **A2** is **not** present anywhere in the lookup array **\$B\$2:\$B\$11**, the **MATCH** function cannot find a position and therefore returns a standard Excel error value: `#N/A`. This error state is the key signal that the value is unique to List A and does not belong in the intersection.

The **ISERROR** function immediately wraps the **MATCH** function: `ISERROR(MATCH(...))`. Its sole purpose is to convert the potentially complex output of **MATCH** (either a numerical position or `#N/A`) into a simple Boolean value (**TRUE** or **FALSE**). If the **MATCH** function successfully finds the item and returns a number, **ISERROR** evaluates to **FALSE** (because no error occurred). Conversely, if **MATCH** returns `#N/A` (indicating failure to find the value), **ISERROR** evaluates to **TRUE**.

The entire structure is governed by the **IF** function, which uses the **ISERROR** result as its logical test. The **IF** syntax is `IF(Logical_Test, Value_if_True, Value_if_False)`. We can observe the two possible outcomes generated by this nested structure:

#### Scenario 1: Value not found (Intersection failure)

The **Logical\_Test** (**ISERROR**) returns **TRUE**.

The **IF** statement executes the `Value_if_True` argument, which is defined as an empty string (`" "`). This results in a blank cell in the output column D.

#### Scenario 2: Value found (Intersection success)

The **Logical\_Test** (**ISERROR**) returns **FALSE**.

The **IF** statement executes the `Value_if_False` argument, which is defined as **A2**. This successfully returns the value from the primary list into the output column D.

By repeating this meticulous process down the length of Column A, the formula systematically identifies and extracts only the elements that exist in both lists, providing a clean, filtered intersection list.

## Extending the Analysis: Handling Duplicates and Alternative Methods

While the IF/ISERROR/MATCH combination is highly effective for identifying the intersection of values, users often encounter complexities related to duplicate values within their lists. If the goal is simply to verify that a match exists, this formula suffices. However, if the lists contain multiple instances of the same item, this method will return every instance found in the primary list (Column A) if it exists at least once in the comparison list (Column B).

For scenarios requiring the removal of duplicates from the final intersection list, a subsequent step is necessary. After the formula has populated Column D, the analyst can utilize Excel's built-in **Remove Duplicates** feature on the results in Column D. This ensures that the final output truly represents a clean set intersection, with each common element appearing only once, aligning perfectly with standard set theory definitions.

## Related Excel Operations for List Management

Understanding how to find the intersection is often a preliminary step to more complex data manipulation tasks. Proficiency in several other common Excel operations significantly enhances list management capabilities. These include functions related to finding unique values, identifying the difference between two lists, and consolidating data from multiple sources. Mastering these related techniques allows for comprehensive data cleanup and comparison:

**Identifying Unique Values:** Utilizing the Advanced Filter feature or newer functions like `UNIQUE` (available in Excel 365) to extract items that appear only once in a list.

**Finding the Set Difference:** Using similar comparison logic, often involving `COUNTIF` or conditional formatting, to highlight or extract items present in List A but absent from List B.

**Conditional Formatting:** Applying conditional formatting based on the `MATCH` or `COUNTIF` functions to visually highlight common elements across the two original lists without creating a separate output column.

The following specialized tutorials offer detailed guidance on how to perform other common list operations and data cleaning tasks within **Excel**: